

---

# **Porter2 Stemmer Documentation**

***Release 1.0***

**Evan Dempsey**

May 30, 2016



<b>1</b>	<b>Porter2 Stemmer</b>	<b>3</b>
1.1	What is stemming? . . . . .	3
1.2	What is it good for? . . . . .	3
1.3	What type of stemmer is this? . . . . .	3
1.4	How do I use it? . . . . .	3
<b>2</b>	<b>Installation</b>	<b>5</b>
<b>3</b>	<b>Usage</b>	<b>7</b>
<b>4</b>	<b>Contributing</b>	<b>9</b>
4.1	Types of Contributions . . . . .	9
4.2	Get Started! . . . . .	10
4.3	Pull Request Guidelines . . . . .	10
4.4	Tips . . . . .	11
<b>5</b>	<b>Credits</b>	<b>13</b>
5.1	Development Lead . . . . .	13
5.2	Contributors . . . . .	13
<b>6</b>	<b>History</b>	<b>15</b>
<b>7</b>	<b>1.0 (2016-03-31)</b>	<b>17</b>
<b>8</b>	<b>Indices and tables</b>	<b>19</b>



Contents:



---

## Porter2 Stemmer

---

An implementation of the Porter2 English stemming algorithm.

- Free software: BSD license
- Documentation: <http://porter2-stemmer.readthedocs.org/>

### 1.1 What is stemming?

Stemming is a technique used in Natural Language Processing to reduce different inflected forms of words to a single invariant root form. The root form is called the stem and may or may not be identical to the morphological root of the word.

### 1.2 What is it good for?

Lots of things, but query expansion in information retrieval is the canonical example. Let's say you are building a search engine. If someone searches for "cat" it would be nice if they were shown documents that contained the word "cats" too. Unless the query and document index are stemmed, that won't happen. Stemming can be thought of as a method to reduce the specificity of queries in order to pull back more relevant results. As such, it involves a trade-off.

### 1.3 What type of stemmer is this?

Porter2 is a suffix-stripping stemmer. It transforms words into stems by applying a deterministic sequence of changes to the final portion of the word. Other stemmers work differently. They may, for instance, simply look up the inflected form in a table and map it to a morphological root, or they may use a clustering approach to map diverse forms to a centre form. Different approaches have different advantages and disadvantages.

### 1.4 How do I use it?

First, install it:

```
$ pip install porter2stemmer
```

Then import it, instantiate a stemmer, and call the stem method with the word you want to stem as an argument:

```
from porter2stemmer import Porter2Stemmer
stemmer = Porter2Stemmer()
print(stemmer.stem('conspicuous'))
```



---

# Installation

---

At the command line:

```
$ easy_install porter2stemmer
```

Or, if you have virtualenvwrapper installed:

```
$ mkvirtualenv porter2stemmer  
$ pip install porter2stemmer
```



---

### Usage

---

To use Porter2 Stemmer in a project:

```
import porter2stemmer
```



---

## Contributing

---

Contributions are welcome, and they are greatly appreciated! Every little bit helps, and credit will always be given. You can contribute in many ways:

### 4.1 Types of Contributions

#### 4.1.1 Report Bugs

Report bugs at <https://github.com/evandempsey/porter2stemmer/issues>.

If you are reporting a bug, please include:

- Your operating system name and version.
- Any details about your local setup that might be helpful in troubleshooting.
- Detailed steps to reproduce the bug.

#### 4.1.2 Fix Bugs

Look through the GitHub issues for bugs. Anything tagged with “bug” is open to whoever wants to implement it.

#### 4.1.3 Implement Features

Look through the GitHub issues for features. Anything tagged with “feature” is open to whoever wants to implement it.

#### 4.1.4 Write Documentation

Porter2 Stemmer could always use more documentation, whether as part of the official Porter2 Stemmer docs, in docstrings, or even on the web in blog posts, articles, and such.

#### 4.1.5 Submit Feedback

The best way to send feedback is to file an issue at <https://github.com/evandempsey/porter2stemmer/issues>.

If you are proposing a feature:

- Explain in detail how it would work.
- Keep the scope as narrow as possible, to make it easier to implement.
- Remember that this is a volunteer-driven project, and that contributions are welcome :)

## 4.2 Get Started!

Ready to contribute? Here's how to set up *porter2stemmer* for local development.

1. Fork the *porter2stemmer* repo on GitHub.
2. Clone your fork locally:

```
$ git clone git@github.com:your_name_here/porter2stemmer.git
```

3. Install your local copy into a virtualenv. Assuming you have virtualenvwrapper installed, this is how you set up your fork for local development:

```
$ mkvirtualenv porter2stemmer
$ cd porter2stemmer/
$ python setup.py develop
```

4. Create a branch for local development:

```
$ git checkout -b name-of-your-bugfix-or-feature
```

Now you can make your changes locally.

5. When you're done making changes, check that your changes pass flake8 and the tests, including testing other Python versions with tox:

```
$ flake8 porter2stemmer tests
$ python setup.py test
$ tox
```

To get flake8 and tox, just pip install them into your virtualenv.

6. Commit your changes and push your branch to GitHub:

```
$ git add .
$ git commit -m "Your detailed description of your changes."
$ git push origin name-of-your-bugfix-or-feature
```

7. Submit a pull request through the GitHub website.

## 4.3 Pull Request Guidelines

Before you submit a pull request, check that it meets these guidelines:

1. The pull request should include tests.
2. If the pull request adds functionality, the docs should be updated. Put your new functionality into a function with a docstring, and add the feature to the list in README.rst.
3. The pull request should work for Python 2.6, 2.7, 3.3, and 3.4, and for PyPy. Check [https://travis-ci.org/evandempsey/porter2stemmer/pull\\_requests](https://travis-ci.org/evandempsey/porter2stemmer/pull_requests) and make sure that the tests pass for all supported Python versions.

## 4.4 Tips

To run a subset of tests:

```
$ python -m unittest tests.test_porter2stemmer
```





---

## Credits

---

### 5.1 Development Lead

- Evan Dempsey <[me@evandempsey.io](mailto:me@evandempsey.io)>

### 5.2 Contributors

None yet. Why not be the first?



---

## History

---



---

**1.0 (2016-03-31)**

---

- First release on PyPI.



---

## Indices and tables

---

- `genindex`
- `modindex`
- `search`